TUGAS

PERANCANGAN SISTEM DIGITAL

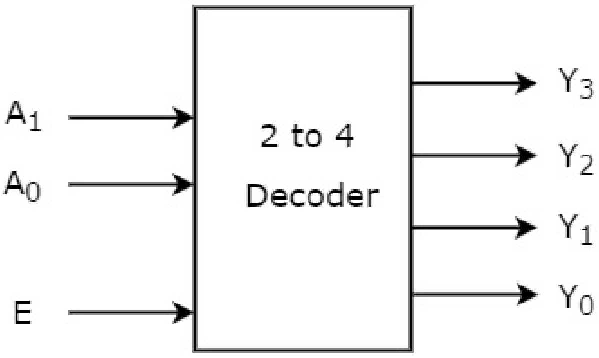
Disusun oleh:
AFRIZAL DANI SAOQI
17/413500/TK/45940

DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA

2020

1 Decoder 2 to 4 using behavioral modeling

1. Architecture



Gambar 1: Architecture

2. Source Code

```
//Initialize verilog module
//stating input and output will be used
module decoder_24(
        input [1:0]w,
        input EN,
        output [3:0]out
);

// Stating behavioral model, output 'out' is stated by 'reg'
// The output is always preserved by the input 'w'
reg [3:0]out;
always @(w,EN)
begin
        if(EN==1'b1) //When Enable
                if (w==2'b00)
                out=4'b0001;
                else if (w==2'b01)
            out=4'b0010;
                else if (w==2'b10)
            out=4'b0100;
```

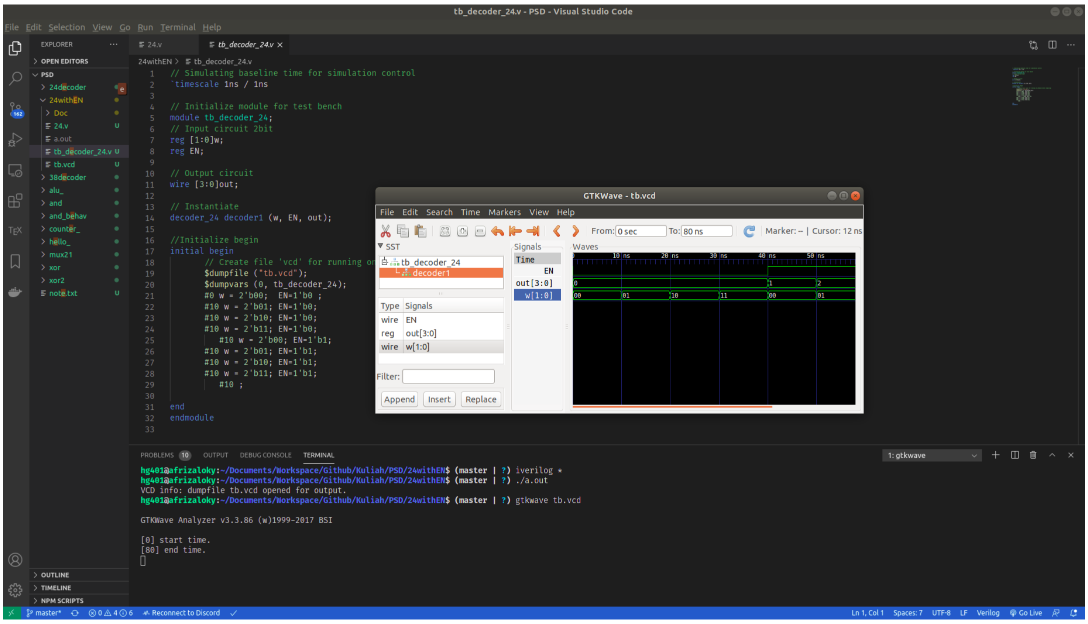
```
                else if (w==2’b11)
            out=4’b1000;
                else
            out=4’bZZZZ; // for rare situation
        else // When disable , output=0
        out=4’b0000;
end
endmodule
```

3. TestBench Source Code

```
// Simulating baseline time for simulation control
‘timescale 1ns / 1ns
// Initialize module for test bench
module tb_decoder_24;
reg [1:0]w; // Input circuit 2bit
reg EN;

// Output circuit
wire [3:0]out; // ouput 4 bit

// Instantiate
decoder_24 decoder1 (w, EN, out);

//Initialize begin
initial begin
        // Create file ’vcd’ for running on gtkwave
        $dumpfile ("tb.vcd");
        $dumpvars (0, tb_decoder_24);
       #0 w = 2’b00;  EN=1’b0 ;
       #10 w = 2’b01; EN=1’b0;
       #10 w = 2’b10; EN=1’b0;
       #10 w = 2’b11; EN=1’b0;
           #10 w = 2’b00; EN=1’b1;
       #10 w = 2’b01; EN=1’b1;
       #10 w = 2’b10; EN=1’b1;
```

```
            #10 w = 2'b11; EN=1'b1;
                #10 ;

    end
    endmodule
```

4. Command Line

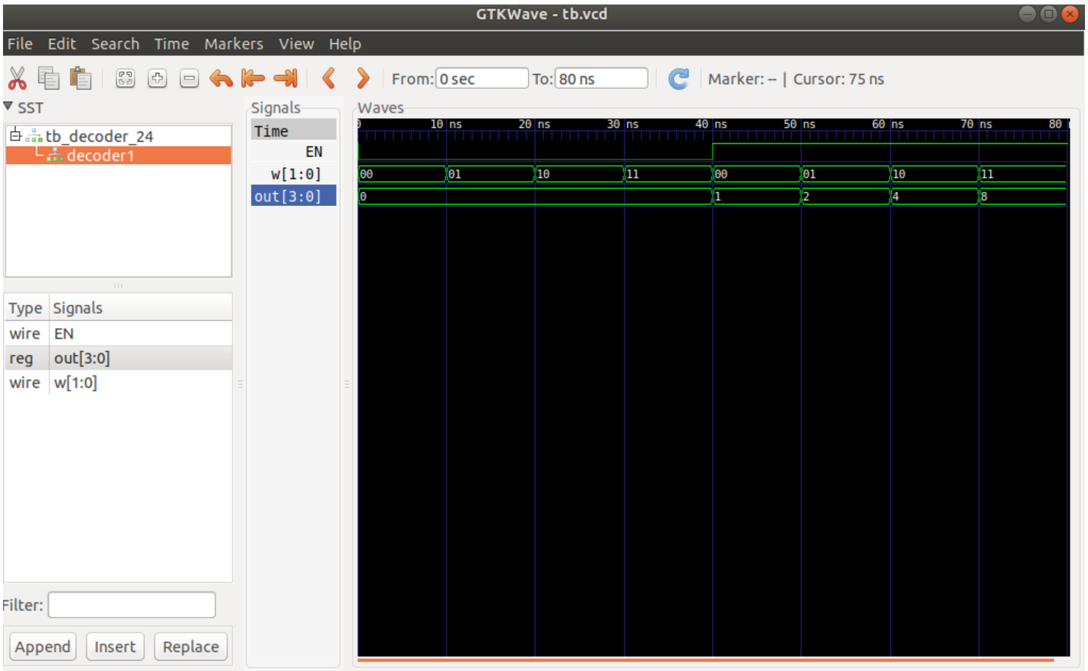


```
// Simulating baseline time for simulation control
`timescale 1ns / 1ns

// Initialize module for test bench
module tb_decoder_24;
// Input circuit 2bit
reg [1:0]w;
reg EN;

// Output circuit
wire [3:0]out;

// Instantiate
decoder_24 decoder1 (w, EN, out);

//Initialize begin
initial begin
        // Create file 'vcd' for running o
        $dumpfile ("tb.vcd");
        $dumpvars (0, tb_decoder_24);
        #0 w = 2'b00;  EN=1'b0 ;
        #10 w = 2'b01; EN=1'b0;
        #10 w = 2'b10; EN=1'b0;
        #10 w = 2'b11; EN=1'b0;
           #10 w = 2'b00; EN=1'b1;
        #10 w = 2'b01; EN=1'b1;
        #10 w = 2'b10; EN=1'b1;
        #10 w = 2'b11; EN=1'b1;
           #10 ;

end
endmodule
```

```
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/24withEN$ (master | ?) iverilog *
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/24withEN$ (master | ?) ./a.out
VCD info: dumpfile tb.vcd opened for output.
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/24withEN$ (master | ?) gtkwave tb.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[80] end time.
```

Gambar 2: Command Line

5. Simulation



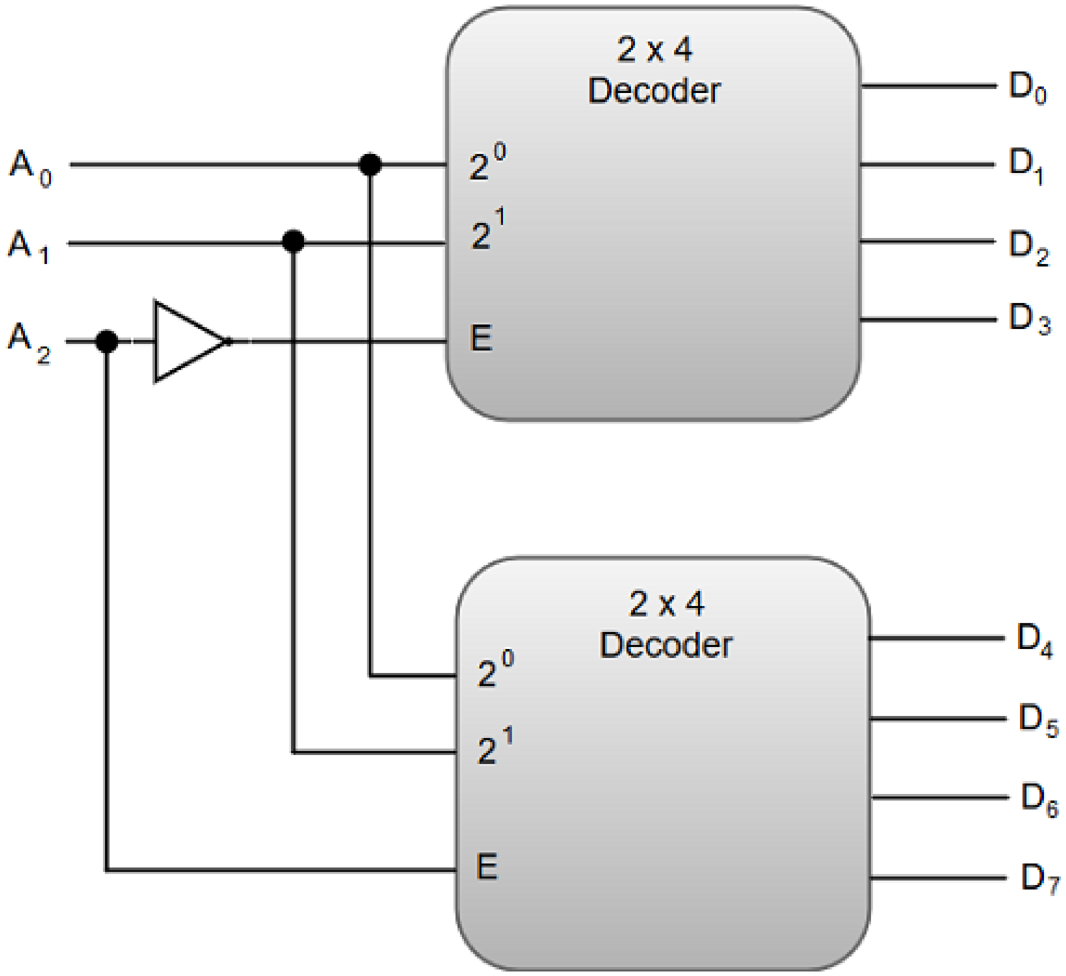
From: 0 sec To: 80 ns Marker: -- | Cursor: 75 ns

Type	Signals
wire	EN
reg	out[3:0]
wire	w[1:0]

Gambar 3: Simulasi Decoder 2 to 4

2 Decoder 3 to 8 using decoder 2 to 4

1. Architecture



Gambar 4: Architecture

2. Source Code 2 to 4

```
//Initialize verilog module
//stating input and output will be used
module decoder_24(
        input [1:0]w,
        input EN,
        output [3:0]out
);

// Stating behavioral model, output ’out’ is stated by ’reg’
// The output is always preserved by the input ’w’
reg [3:0]out;
```

```
always @(w,EN)
begin
        if(EN==1'b1) //When Enable
                if (w==2'b00)
                out=4'b0001;
                else if (w==2'b01)
            out=4'b0010;
                else if (w==2'b10)
            out=4'b0100;
                else if (w==2'b11)
            out=4'b1000;
                else
            out=4'bZZZZ; // for rare situation
        else // When disable , output=0
        out=4'b0000;
end
endmodule
```

3. Source Code 3 to 8 using 2 to 4

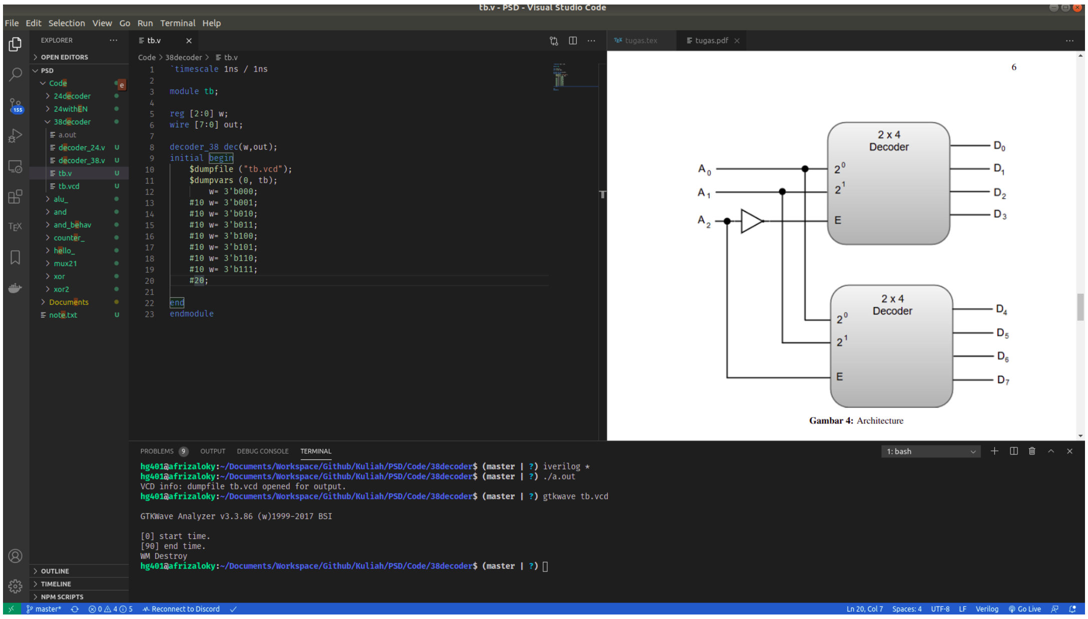
```
module decoder_38 (
    input [2:0]w, // declaration 3 bit input
        output [7:0]out // declaration 8 bit output
);

 // instance module decoder 2 to 4
 decoder_24 dec0 ( w[1:0], ~w[2],  out[3:0]);
 decoder_24 dec1 ( w[1:0], w[2] ,  out[7:4]);

endmodule
```

4. TestBench Source Code

```
// Simulating baseline time for simulation control
`timescale 1ns / 1ns

// Initialize module for test bench
```

```
module tb;

reg [2:0] w; // input 3 bit
wire [7:0] out; // output 8 bit

decoder_38 dec(w,out); // instantiate
initial begin
    $dumpfile ("tb.vcd");
        $dumpvars (0, tb);
        w= 3'b000;
    #10 w= 3'b001;
    #10 w= 3'b010;
    #10 w= 3'b011;
    #10 w= 3'b100;
    #20 w= 3'b101;
    #20 w= 3'b110;
    #20 w= 3'b111;
    #20;

end
endmodule
```

5. Command Line

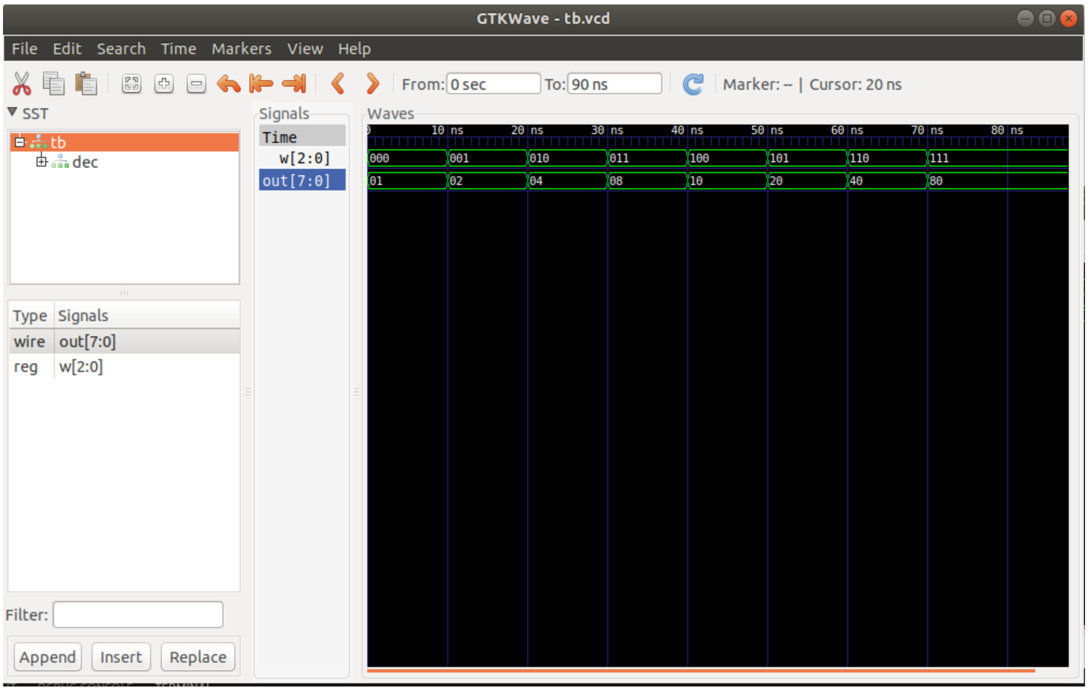


```
`timescale 1ns / 1ns

module tb;

reg [2:0] w;
wire [7:0] out;

decoder_38 dec(w,out);
initial begin
    $dumpfile ("tb.vcd");
    $dumpvars (0, tb);
        w= 3'b000;
    #10 w= 3'b001;
    #10 w= 3'b010;
    #10 w= 3'b011;
    #10 w= 3'b100;
    #10 w= 3'b101;
    #10 w= 3'b110;
    #10 w= 3'b111;
    #20;

end
endmodule
```

```
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/Code/38decoder$ (master | ?) iverilog *
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/Code/38decoder$ (master | ?) ./a.out
VCD info: dumpfile tb.vcd opened for output.
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/Code/38decoder$ (master | ?) gtkwave tb.vcd

GTKWave Analyzer v3.3.86 (w)1999-2017 BSI

[0] start time.
[90] end time.
WM Destroy
hg401@afrizaloky:~/Documents/Workspace/Github/Kuliah/PSD/Code/38decoder$ (master | ?) 
```

Gambar 5: Command Line

6. Simulation



```
Type  Signals
wire  out[7:0]
reg   w[2:0]
```

```
Time       0    10 ns   20 ns   30 ns   40 ns   50 ns   60 ns   70 ns   80 ns
w[2:0]     000  001     010     011     100     101     110     111
out[7:0]   01   02      04      08      10      20      40      80
```

Gambar 6: Simulasi Decoder 3 to 8

Terdapat perbedaan antara hasil Simulasi dengan perhitungan manual.

```
decoder_24 dec0 ( w[1:0], ~w[2],  out[3:0]);
decoder_24 dec1 ( w[1:0], w[2] ,  out[7:4]);
```

Jika input 100_2, menurut potongan kode di atas
$w_0 = 0, w_1 = 0, w_2 = 1$. Output dari dec0 adalah 0000_2 karena EN=0. Output dari dec1 adalah 0001_2 karena EN=1 dengan input = 00_2. Jika kedua output tersebut digabungkan maka output=$00010000_2 = 16_{10}$

Pada perhitungan manual :

$Input$	0	1	2	3	4	5	6	7
$Output$	1	2	4	8	16	32	64	128

Tabel 1: perhitungan manual

Pada simulasi:

$Input$	0	1	2	3	4	5	6	7
$Output$	1	2	4	8	10	20	40	80

Tabel 2: Hasil simulasi

3 Repository

Github Repository